

## Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

### Listing of Claims:

1. (Currently amended) A method for mapping components of an XML schema using a program, comprising:
  - a) receiving XML schema components;
  - ~~a:~~ b) uniquely mapping the XML schema components to ~~with~~ a conversion language; and
  - ~~b:~~ c) receiving unique names for uniquely naming components of the conversion language based on names of the XML schema components; and  
wherein step c) further comprises at least one of the group consisting of: (1) hashing the name of each element having the same name and type as another element to form an element name hash code part of a Java member variable, (2) hashing the name of each attribute having the same name as another attribute to form an attribute name hash code part of a further Java member variable, (3) hashing a QName for each complex type to form a complex type name hash code part of a Java class name, (4) hashing each LongName name to form a LongName hash code part of a truncated Java name, and (5) hashing a concatenated string of all component names of an anonymous complex type component to form an anonymous complex type hash code part of a further Java class name.
2. (Currently amended) The method of claim 1, wherein the conversion language is Java, and step ~~a:~~ b) further comprises steps of (1) mapping each XML schema element and type to a Java component and (2) uniquely identifying each XML schema element and type within a set of all distinct XML schema.

3. (Currently amended) The method of claim 1, wherein the conversion language is Java, and step ~~b~~: c) further comprises, for each different XML schema element and type, generating a unique Java component name.

4. (Original) The method of claim 3, wherein the step of generating each said unique Java component name comprises generating each name so that each said name substantially adheres to Java naming standards, and so that each said name remains the same in subsequent mappings of XML schema components when an XML schema component label on which said name is based remains the same.

5. (Original) The method of claim 1: wherein the conversion language is Java; step a. further comprises (1) mapping each XML schema element, type and attribute to a Java component and (2) uniquely identifying each XML schema element, type and attribute within a set of all distinct XML schema; and step b. further comprises, for each different XML schema element, type and attribute name, generating a unique Java component name; and the method further comprises generating a reusable definition object operable in converting information between an XML object associated with the XML schema and a Java object.

6. (Cancelled)

7. (Currently amended) The method of claim 1 ~~6~~, further comprising appending a suffix to a generated Java component name based on a first XML schema component name when the generated Java component name is identical to a previously generated Java component name based on a second XML schema component name different from the first XML schema component name.

8. (Currently amended) An information handling system comprising a processor and an object definition tool for generating an object operable in mapping components of an XML schema, the object definition tool comprising plural instructions and the processor is operably configured to execute said plural instructions, the plural instructions comprising: a. mapping instructions operable for uniquely mapping XML schema components with a conversion language; and b. naming instructions operable for uniquely naming components of the conversion language based on names of the XML schema components;

wherein the naming instructions further comprise hashing instructions configured to perform at least one of the group of operations consisting of: (1) hashing the name of each element having the same name and type as another element to form an element name hash code part of a Java member variable, (2) hashing the name of each attribute having the same name as another attribute to form an attribute name hash code part of a further Java member variable, (3) hashing a QName for each complex type to form a complex type name hash code part of a Java class name, (4) hashing each LongName name to form a LongName hash code part of a truncated Java name, and (5) hashing a concatenated string of all component names of an anonymous complex type component to form an anonymous complex type hash code part of a further Java class name.

9. (Original) The information handling system of claim 8, wherein the conversion language is Java, and the mapping instructions comprise instructions configured to (1) map each XML schema element and type to a Java component and (2) uniquely identify each XML schema element and type within a set of all distinct XML schema.

10. (Original) The information handling system of claim 8, wherein the conversion language is Java, and the naming instructions further comprise instructions, for each different XML schema element and type, configured to generate a unique Java component name.

11. (Original) The information handling system of claim 10, wherein naming instructions comprise yet further instructions configured to generate each name so that each said

name substantially adheres to Java naming standards, and so that each said name remains the same in subsequent mappings of XML schema components when an XML schema component label on which said name is based remains the same.

12. (Original) The information handling system of claim 8: wherein the conversion language is Java; the mapping instructions comprise instructions configured to (1) map each XML schema element and type to a Java component and (2) uniquely identify each XML schema element and type within a set of all distinct XML schema; and the naming instructions further comprise additional instructions configured to generate, for each different XML schema element and type, a unique Java component name; the system further comprising object definition instructions configured to generate a reusable definition object operable in converting information between an XML object associated with the XML schema and a Java object.

13. (Cancelled)

14. (Currently amended) The system of claim 12 ~~13~~, further comprising appending instructions operable for appending a suffix to a generated Java component name based on a first XML schema component name when the generated Java component name is identical to a previously generated Java component name based on a second XML schema component name different from the first XML schema component name.

15. (Original) The system of claim 12, further comprising an adapter configured to operably convert XML objects using the reusable definition object to converted XML objects, and an application and computer operable together and configured to receive and perform operations on the converted XML objects.

16. (Currently amended) A program product in a signal bearing medium executable by a device for generating an object operable in mapping components of an XML schema, the product comprising: a. mapping instructions operable for uniquely mapping XML schema components with a conversion language; and b. naming instructions operable for uniquely naming components of the conversion language based on names of the XML schema components;

wherein the naming instructions further comprise hashing instructions configured to perform at least one of the group of operations consisting of: (1) hashing the name of each element having the same name and type as another element to form an element name hash code part of a Java member variable, (2) hashing the name of each attribute having the same name as another attribute to form an attribute name hash code part of a further Java member variable, (3) hashing a QName for each complex type to form a complex type name hash code part of a Java class name, (4) hashing each LongName name to form a LongName hash code part of a truncated Java name, and (5) hashing a concatenated string of all component names of an anonymous complex type component to form an anonymous complex type hash code part of a further Java class name.

17. (Original) The program product of claim 16, wherein the conversion language is Java, and the mapping instructions comprise instructions configured to (1) map each XML schema element and type to a Java component and (2) uniquely identify each XML schema element and type within a set of all distinct XML schema.

18. (Original) The program product of claim 16, wherein the conversion language is Java, and the naming instructions further comprise instructions, for each different XML schema element and type, configured to generate a unique Java component name.

19. (Original) The program product of claim 18, wherein naming instructions comprise yet further instructions configured to generate each name so that each said name substantially adheres to Java naming standards, and so that each said name remains the same in subsequent mappings of XML schema components when an XML schema component label on which said name is based remains the same.

20. (Original) The program product of claim 16: wherein the conversion language is Java; the mapping instructions comprise instructions configured to (1) map each XML schema element and type to a Java component and (2) uniquely identify each XML schema element and type within a set of all distinct XML schema; and the naming instructions further comprise additional instructions configured to generate, for each different XML schema element and type, a unique Java component name; the product further comprising object definition instructions configured to generate a reusable definition object operable in converting information between an XML object associated with the XML schema and a Java object.

21. (Canceled)

22. (Currently amended) The program product of claim 20 ~~21~~, further comprising appending instructions operable for appending a suffix to a generated Java component name based on a first XML schema component name when the generated Java component name is identical to a previously generated Java component name based on a second XML schema component name different from the first XML schema component name.

23. (Original) The program product of claim 22, further comprising adapter instructions configured to operably convert XML objects using the reusable definition object to converted XML objects.